Frugal Source-Grounded Detection of Graded Omissions in LLM-Generated French Medical Summaries

Hodhaifa Benouaklil[1,2] **Achir Oukelmoun**[3] **Nasredine Semmar**[1]
Faiza Belbachir[4] Jaafer Klila[1] Hassane Essafi[1]

[1]Université Paris-Saclay, CEA, List, 91120 Palaiseau, France
[2]Université Grenoble Alpes, 38000 Grenoble, France
[3]Université Paris-Saclay, 91190 Gif-sur-Yvette, France
[4]LyRIDS, ECE Research Center, 75015 Paris, France

Abstract

Large language models can produce fluent clinical summaries whose most dangerous errors are often omissions: clinically relevant facts may disappear without creating an explicit contradiction. In privacy-constrained hospital settings, an omission detector must run on-premise, require no gold summary, and avoid additional large language model calls for evaluation. We reproduce and stress-test EmbedKDECheck, a source-grounded, reference-free embedding-density detector, on a controlled French medical corpus in which summaries are generated from source variants with known fact-deletion rates across sixteen graded levels. We adapt the published pipeline to this graded setting with a bounded mean-based score, and replace principal component analysis with a random projection. The resulting configuration improves all reported held-out metrics while reducing the dimensionality-reduction cost, reaching a Spearman correlation of 0.48, F1 of 0.80 on complete-versus-half-omitted summaries, and roughly 573 thousand floating-point operations per document pair. We then characterise the main remaining limitation: a report-specific score offset that explains most score variance and limits corpus-level rank correlation. Targeted follow-up interventions show that this offset is removable only with anchor summaries that violate the frugality constraint. Overall, the simple static-embedding and random-projection pipeline offers a frugal operating point for source-grounded completeness evaluation, lying on the graded rank-correlation/cost frontier among non-LLM and mid-cost reference-free metrics.

1 Introduction

Large language models (LLMs) are increasingly considered for drafting clinical documentation, including discharge notes, imaging reports, and referral letters. These are summarisation tasks, and current models can produce fluent, plausible summaries. The failure mode that worries clinicians is not only hallucination: fabricated findings may at least create contradictions with the source record. Omissions are more silent. A summary that leaves out a nodule measurement, an allergy, or a contraindication may be shorter, cleaner, and entirely plausible, with no surface signal that critical information is missing. This issue is distinct from factual fabrication and has received less attention than hallucination or factual inconsistency in the evaluation literature (Oukelmoun et al., 2025a; Luo et al., 2024).

Detecting omissions automatically is therefore a guard-rail problem, but the guard-rail carries constraints of its own. In privacy-constrained hospital settings, clinical text contains health data subject to strict European and French data-protection requirements, including GDPR restrictions on health-data processing and certified hosting requirements for personal health data in France (European Parliament and Council of the European Union, 2016; République Française, 2026). This makes local or certified on-premise processing preferable to commercial API-based judges. The screening must also run over every generated summary, which motivates a low incremental evaluation cost. We treat summary generation as an upstream process shared by all evaluated metrics and focus on the additional cost of checking an already-generated summary. What is needed is therefore a frugal, reference-free detector: no additional LLM inference for evaluation, no gold summary to compare against, only the source report and the candidate summary.

EmbedKDECheck (Oukelmoun et al., 2025a) is a natural starting point under these constraints. It embeds the tokens of the report and of the summary into a low-dimensional space and asks, for each report token, how plausible that token is under a density model of the summary. Report content that the summary does not cover should sit in low-density regions of that model, and the per-token plausibilities are aggregated into an omission score. No

Proceedings of the 19th International Natural Language Generation Conference: Long Papers, pages 1383–1395
October 17-21, 2026 ©2026 Association for Computational Linguistics

transformer is involved in the detector at inference time. The method was introduced and validated on a binary task: a complete summary versus one missing roughly half of the source sentences. This raises two practical questions: whether the method transfers to graded omission, where severity varies continuously across deletion rates rather than being present or absent, and what limits its performance once tuned.

This paper is a systematic reproduction and stress-test of EmbedKDECheck for graded omission detection in French medical summaries. We show that two modifications, a bounded mean aggregation and a random projection, improve the original frugal pipeline, while exposing a report-specific score offset that limits corpus-level rank correlation. Concretely, our contributions are:

- a controlled-omission corpus of French clinical reports in which summaries are generated from source variants with known fact-deletion rates (Section 3);
- two adaptations of the published pipeline: a bounded sigmoid-mean score for the graded setting and a random projection that outperforms principal component analysis on held-out metrics while reducing reduction cost (Section 4);
- a diagnosis of the method's structural noise ceiling, a per-report additive offset, and a set of targeted interventions showing which offset-removal strategies fail under the frugal deployment constraint (Section 5);
- a comparison against established source-grounded and reference-free metrics under the same on-premise, no-gold-summary constraint, showing that the simple pipeline lies on the graded rank-correlation/cost frontier among frugal and mid-cost alternatives (Section 6).

2 Background and Related Work

Automatic evaluation of generated summaries has moved beyond reference-based overlap toward source-grounded and reference-free evaluation. This distinction is central for omission detection. Factual inconsistency metrics usually ask whether a summary contains unsupported claims, whereas omission detection asks the complementary question: whether source content is unsupported by the summary. Existing approaches include QA-based metrics such as QAGS (Wang et al., 2020) and QAFactEval (Fabbri et al., 2022), entailment-based metrics such as SummaC (Laban et al., 2022), embedding-based coverage metrics such as BERTScore (Zhang et al., 2020), and LLM-based judges such as SelfCheckGPT (Manakul et al., 2023), G-Eval (Liu et al., 2023), GPTScore (Fu et al., 2024), and ChainPoll (Friel and Sanyal, 2023). These methods can be effective, but many require transformers, question-generation pipelines, or repeated LLM inference, which is difficult to reconcile with low-cost on-premise screening in privacy-constrained clinical settings.

EmbedKDECheck (Oukelmoun et al., 2025a,b), the method studied here, occupies a lighter point in this design space. Its core idea is to measure how well each source token is covered by the semantic regions represented in the candidate summary. Given a source report x_i and a candidate summary x_o, it tokenises both texts and maps each token to a 300-dimensional FTW2V embedding (Oukelmoun et al., 2023), a French medical word embedding combining Word2Vec (Mikolov et al., 2013) and FastText (Bojanowski et al., 2017). The report and summary token clouds are jointly standardised and reduced with PCA. A Gaussian KDE (Węglarczyk, 2018; Silverman, 1986) is then fitted on the reduced summary tokens, and each reduced report token t is evaluated under this summary density through the ratio

$$r(t) = \frac{\hat{f}_o(t)}{\min_{u \in x_o} \hat{f}_o(u)}, \tag{1}$$

where $\hat{f}_o$ is the KDE fitted to the summary-token embeddings. Source tokens that are semantically well represented in the summary should lie in high-density regions and receive large ratios, whereas poorly covered or omitted source tokens should receive small ratios.

The original method converts these token-level coverage signals into a document-level omission score using the maximum, $\max_t 1/r(t)$, making the final score depend on the single worst-covered report token. It was evaluated mainly on a binary complete-versus-omitted setting. Our work keeps the reference-free, non-LLM structure of the method but adapts it to graded omission severity by replacing the max aggregation with a bounded mean-based score and by replacing per-pair PCA with a fixed random projection.

3 A Controlled-Omission Corpus

An omission detector cannot be measured without knowing what was omitted. Naturally occurring LLM summaries do omit content, but the missing

content and its amount are usually unknown. We therefore built a corpus in which omission pressure is injected by construction, so that every generated summary is associated with a known source fact-deletion rate (Fig. 1). The key design choice is to delete facts from the report *before* summarising it: the summariser is asked to summarise faithfully a text that is already missing content. This makes the injected omission largely exogenous. The summariser may still drop further content of its own, so the deletion rate is not the exact amount missing from the final summary; it is, however, a known and monotonically increasing driver of omission, and a faithful detector's score is expected to rise with it. The resulting graded benchmark is publicly available on Hugging Face (Benouaklil et al., 2026).

The source documents are 460 French synthetic clinical reports, drawn from the 674-report EmbedKDECheck corpus (Oukelmoun et al., 2025a). That corpus was generated from 50 real, anonymised seed reports that were modified into fictitious seeds and expanded with GPT-4-32K (Achiam et al., 2023) while preserving clinical structure. Its realism was validated by expert surgeons: human-versus-synthetic discrimination was near chance, while omission labels reached 95% accuracy and Cohen's $\kappa = 0.82$ on the manually reviewed subset (Oukelmoun et al., 2025a). Using these reports keeps our setting comparable to the one for which EmbedKDECheck was designed, while allowing us to construct a graded version of the task.

The unit of omission is the atomic fact. Each report is decomposed by an LLM into a numbered list of self-contained factual statements, with section titles preserved as structural markers. The choice of segmenter was made empirically: six open-weight models (Llama 3 70B and Llama 3.1 8B (Dubey et al., 2024), Mistral Small 24B (Mistral AI, 2025), Ministral 8B (Mistral AI, 2024), Qwen3 30B (Yang et al., 2025), and Gemma 4 31B (Google DeepMind, 2026)) each segmented all 460 reports, and the outputs were compared on reference-free string metrics (Appendix A). Coverage is ROUGE-1 recall of the concatenated facts against the source, and faithfulness is ROUGE-1 precision. Together with the number of extracted facts, these diagnostics characterise content preservation and segmentation granularity. Gemma 4 31B was retained because it combined perfect faithfulness with the finest fact granularity, extracting 46.2 facts per report on average (median 37; Appendix A), which matters because each fact is the smallest deletable unit. Using a single segmenter throughout also keeps fact counts comparable across the corpus.

The deletion respects document structure: facts are grouped under their section titles, and each section keeps at least one fact, so no section disappears entirely. After deletion, numeric prefixes and the artificial section markers introduced during segmentation are removed, and the surviving section titles and facts are concatenated in their original order as newline-separated text for summarisation. Within a section, the deletion order is fixed per report, so the facts removed at 20% are a subset of those removed at 40%. The realised deletion seeds are distributed with the public dataset, enabling exact replay of the released variants. Because rounding and the keep-one-per-section constraint make the achieved rate differ slightly from the target, all evaluations use the achieved deletion rate.

Each omission-injected report is then summarised by four open-weight models: Gemma 4 31B, Llama 3 70B, Mistral Small 24B, and Qwen3 30B. All models use a common instruction requiring a faithful clinical summary of bounded length with no added information. The summarisation, segmentation, and anchor-generation prompts are listed in Appendix B; English translations are provided for readability only. The main detection experiments use Gemma 4 31B for both segmentation and summarisation, keeping the corpus construction internally consistent. The other models' summaries are retained for the cross-model analyses of Section 5.

At evaluation time, the detector receives the original full report as the source document and the summary generated from the omission-injected variant as the candidate summary. The achieved deletion rate of the variant is used as the controlled omission-severity label. Thus, the deletion rate is a monotonic proxy for missing source information rather than a direct annotation of the final summary's exact omission count.

4 Pipeline Adaptation and Evaluation Protocol

4.1 Score adaptation for graded omission

The original EmbedKDECheck aggregation uses the maximum token score, which is natural for a binary alarm: one badly covered source token is enough to flag a summary. For graded omission, however, we want the score to reflect how much

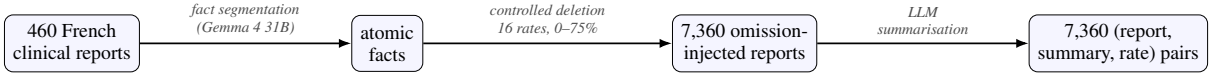


Figure 1: Controlled-omission corpus construction. Facts are deleted before summarisation; evaluation compares each generated summary with the original full report and uses achieved deletion rate as the severity label.

source content is missing, not only the worst individual token. The max rule is therefore unstable because a single routine outlier can dominate the score of the entire report-summary pair. On our corpus, at deletion rate 0, where scores should be low, the 95th percentile of the original max-based scores was around 142,000. The cause is not necessarily omission: a tokenisation artefact or rare word can fall in a very low-density region of the summary cloud, making $1/r(t)$ explode.

We therefore replace the max aggregation with a bounded mean. For each report token t, we convert the density ratio of (1) into a local omission score $\ell(t) = \sigma(1/r(t))$, where σ is the logistic sigmoid. Since covered tokens have large $r(t)$ and omitted or poorly covered tokens have small $r(t)$, $\ell(t)$ increases with local under-coverage while remaining bounded. The document score is then

$$S(x_i, x_o) = \frac{1}{|T_i|} \sum_{t \in T_i} \ell(t), \tag{2}$$

where T_i is the set of retained source tokens. This score lies in $(0.5, 1]$, is robust to single-token outliers, and remains responsive across the omission range where the max aggregation saturates. Hyperparameter tuning later confirmed the mean as the best aggregation among the six candidates we evaluated.

4.2 Evaluation protocol

Unless explicitly noted, all headline numbers follow the same held-out protocol. The 460 reports are split 80/20 by report identifier with seed 0: 368 reports for training and 92 for testing, yielding 5,752 training pairs and 1,424 test pairs. Of the 7,360 generated report–summary pairs, 184 (2.5%) were excluded by a prespecified minimum-length criterion because the summary contained at most five retained content tokens after preprocessing; no source report failed this criterion. All rate variants of a given synthetic report are assigned to the same side of the split. Since the 460 synthetic reports originate from 50 seed reports, reports derived from the same seed may occur across partitions; however, EmbedKDECheck is training-free, so the split is used only for hyperparameter selection and threshold calibration rather than model fitting. Because each retained test report contributes the same rate ladder, the rate distribution is uniform by construction up to the minimum-length filtering.

We report four metrics. Spearman's ρ between the detector score and the achieved fact-deletion rate is the main graded metric. F1@50 and AUC@50 measure the binary task "complete versus half omitted", matching the original complete-versus-omitted setting. Finally, mean F1 averages the fifteen binary F1 scores obtained by separating complete summaries from each non-zero deletion level, from 5% to 75%.

F1@50 must be interpreted with care because the complete-versus-half-omitted task is roughly balanced: an always-positive classifier would already obtain F1 ≈ 0.634. We therefore treat ρ and AUC@50 as the primary discrimination metrics and report F1 values as thresholded deployment summaries.

Hyperparameters are selected on the training split only, and the selected configuration is evaluated once on the held-out test split. Decision thresholds are also learned on the training split and frozen before test evaluation, which matches deployment: a production system has no test labels on which to retune its cutoff. Confidence intervals are computed with a clustered bootstrap over reports, not over pairs, because a report's rate variants are correlated. Specifically, we resample the 92 test reports with replacement $B = 2000$ times and recompute each metric. For pairwise comparisons, a paired clustered bootstrap recomputes both methods on the same resampled reports, yielding the difference Δ and the probability $P(\Delta > 0)$ used in significance claims. The threshold-free metrics, ρ and AUC@50, are the primary discrimination metrics; F1 values are reported as deployment-relevant thresholded summaries.[1]

[1]The lever experiments of Section 5 use a lighter evaluation harness in which the F1 threshold is selected within the evaluation split, whereas the headline protocol selects the threshold on the training split and freezes it before test evaluation. Their absolute F1@50 values can therefore be optimistic relative to the headline value; lever results are interpreted primarily through within-lever deltas and threshold-free ρ.

4.3 Reducer selection

The KDE relies directly on distances between token embeddings, while density estimation becomes less reliable and more expensive in the original high-dimensional embedding space. EmbedKDECheck therefore reduces the joint report-summary token cloud before fitting the KDE; the original method uses PCA for this step. We treat this dimensionality reducer as a tunable component and evaluate each candidate under the same train/test protocol. Table 1 reports the best configuration for each reducer, selected on the training split and evaluated once on the held-out reports. Random projection (Achlioptas, 2003) is the clear winner: it improves all four held-out metrics relative to PCA while also reducing the dimensionality-reduction cost by roughly a factor of six, since the projection matrix is drawn once and reused whereas PCA must fit a local subspace for each report-summary pair. The gain over the runner-up PCA is statistically significant under a paired clustered bootstrap ($\Delta\rho = +0.094$, 95% CI $[+0.050, +0.142]$, $P = 1.00$). To verify that this advantage is not an artefact of a single fortunate projection, we redrew the random-projection matrix under $K = 10$ seeds and re-evaluated on the held-out split. Performance is stable across seeds (mean $\pm$ standard deviation): $\rho = 0.470 \pm 0.015$, AUC@50 $= 0.861 \pm 0.012$, F1@50 $= 0.794 \pm 0.032$, and mean F1 $= 0.658 \pm 0.015$. The reported seed-0 configuration lies within about one standard deviation of the seed mean on ρ, F1@50, and mean F1, and modestly above it on AUC@50; in every case the seed-to-seed spread is far smaller than the $+0.094$ margin over PCA, so the random-projection advantage is not seed-dependent.

This advantage is plausibly explained by stability: PCA is fitted separately on each small report-summary token cloud, whereas random projection uses a single fixed matrix for all pairs. This avoids per-pair subspace variability, and in our diagnostics it also distorted token-frequency bands more evenly than PCA, which tended to compress rare-token distances more strongly. Additional reducer-geometry diagnostics, including token-frequency distortion, are reported in Appendix C.

The nonlinear reducers were included to test whether a more flexible geometry could improve the KDE space. They do not. UMAP (McInnes et al., 2018) reaches only $\rho = 0.215$, and t-SNE (van der Maaten and Hinton, 2008) collapses to a constant-predictor point, making ρ undefined and AUC@50 equal to 0.500. This is consistent with the role of the reducer in EmbedKDECheck: the KDE relies on meaningful local distances, whereas visualisation-oriented nonlinear reducers can warp distances in ways that are useful for display but harmful for density estimation. In our sweep, these nonlinear methods were also much slower because they require a new fit for each document pair, unlike the fixed random-projection baseline.

Table 1: Reducer ablation on the held-out split.

reducer	bw	dim	agg	ρ	F1@50	AUC@50	meanF1
random projection	3.0	3	mean	**0.479**	**0.796**	**0.881**	**0.656**
PCA	3.5	10	q3	0.385	0.739	0.818	0.601
kernel PCA	0.7	7	median	0.068	0.646	0.563	0.529
UMAP	0.7	5	p95	0.215	0.638	0.676	0.544
t-SNE	0.7	3	max	–	0.649	0.500	0.529

4.4 Held-out performance

The selected configuration is the random-projection row of Table 1: FTW2V with a random projection to three dimensions, bandwidth 3.0, sigmoid-mean aggregation, and a decision threshold selected on the training split. On the 1,424 held-out pairs, it reaches $\rho = 0.479$ (95% CI $[0.40, 0.55]$), AUC@50 $= 0.881$ ($[0.83, 0.92]$), F1@50 $= 0.796$ ($[0.750, 0.838]$), and mean F1 $= 0.656$ ($[0.627, 0.681]$), with clustered-bootstrap intervals over test reports. As shown in Fig. 2, discrimination improves with omission severity, with per-rate AUC rising from 0.66 at 5% deletion to 0.97 at 75%.

We also tested contextual French biomedical token embeddings. The best CamemBERT-bio (Touchent and Villemonte de la Clergerie, 2024; Martin et al., 2020) configuration reached only $\rho = 0.353$, and DrBERT (Labrak et al., 2023) reached $\rho = 0.298$, both below the static FTW2V baseline. Running KDE directly in the unreduced high-dimensional embedding space collapsed all three embedders to chance-level AUC@50, consistent with distance concentration in high dimension (Beyer et al., 1999; Aggarwal et al., 2001). These results suggest that the main gain comes from a stable low-dimensional density space rather than from contextual token representations.

5 The Per-Report Offset and Six Levers

5.1 Diagnosis: a per-report offset

The main limitation of the tuned detector is revealed by a variance decomposition. A one-way analysis of variance with the omission rate as the

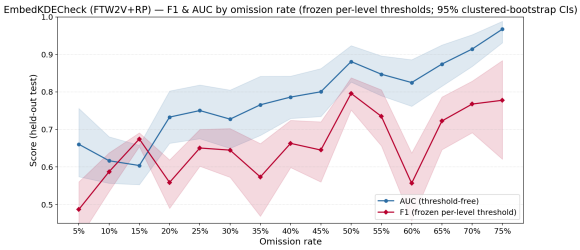


Figure 2: Held-out AUC@50 and F1 by deletion rate, with clustered-bootstrap confidence intervals.

grouping factor attributes about 75% of the score variance to within-rate variation, that is, to differences between reports at the same omission rate, and only about 25% to the omission rate itself. Two summaries at the same 25% omission rate can therefore receive very different scores, for example 0.70 and 0.90, depending mainly on which report they summarise. Fig. 3 shows both views of this effect: individual report curves generally rise with the omission rate but sit at report-specific heights, while pooled per-rate distributions overlap heavily.

The bottleneck is across-report calibration, not within-report ordering. If we restrict attention to a single report and rank its sixteen variants by score, the detector performs well: the median within-report Spearman correlation is $+0.76$, and 85% of reports have a positive within-report correlation. The scores for each report therefore behave as if a report-specific constant were added to all variants of that report. As an oracle diagnostic, we subtract from every score the mean score of its own report. This operation is not deployable, because it uses the full rate ladder of the report, but it estimates the gain available from perfect offset removal. It raises corpus-level Spearman correlation from 0.479 to 0.605, a gain of about $+0.13$.

We then tested whether this offset can be predicted from the report itself. We regressed the oracle offset on seven scalar properties of the report-summary token clouds that are computable at inference time, including the number of report tokens, the report-cloud spread (mean pairwise distance), self-density (mean KDE density of report tokens under a KDE fitted on the report itself), summary-cloud size, and report length. The best single proxy correlates with the oracle offset at only $|\rho| = 0.127$, and combining all seven raises the correlation only to 0.154. Thus, the report's gross embedding geometry does not explain its score level. Removing the offset appears to require information beyond a single isolated report-summary pair.

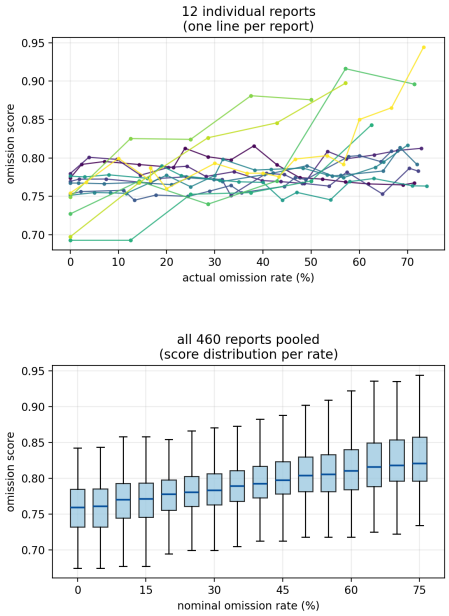


Figure 3: Per-report offset shown in two complementary views. Top: score curves for 12 individual reports across omission rates. Bottom: pooled score distributions across all 460 reports by omission rate.

5.2 Lever 1: removing the offset with anchor summaries

If each report adds a roughly constant shift to its scores, that shift can in principle be estimated and subtracted. We test this idea with *anchor summaries*: additional summaries of the same report used only to estimate the report-specific score level. Each anchor is scored by the same EmbedKDECheck pipeline, and the audited summary's corrected score is its raw score minus the mean anchor score. Anchor batch size varies by scheme: Schemes 1, 2, and 5 use $k = 5$ anchors, Scheme 3 uses $k = 6$ rate-spanning anchors, and Scheme 4 evaluates $k \in \{1, 2, 3\}$ same-rate anchors. Writing $s(\cdot)$ for the raw omission score and $\mathcal{A}_r = \{a_1, \ldots, a_k\}$ for the anchor batch of report r, the corrected score of an audited summary x is

$$\tilde{s}(x) = s(x) - \frac{1}{k} \sum_{a \in \mathcal{A}_r} s(a). \tag{3}$$

The quality of this estimate depends on three properties of the anchor batch. First, the anchors should be generated by the same model as the audited summary. Second, they should use the same prompt. Third, their omission rates should span a range rather than concentrate at one level. The first two properties matter because the offset is tied to the report–model–prompt combination; anchors generated under a different model or prompt estimate

Table 2: Anchor-based offset removal. Baseline $\rho = 0.479$; oracle demeaning reaches $\rho = 0.605$. Differences are computed from unrounded values.

scheme	same model	same prompt	rate spread	ρ	$\Delta\rho$
1	✓	✓	✓	**0.580**	$+0.10$
2	✓	×	✓	0.470	-0.009
3	×	✓	✓	0.473	-0.007
4	×	✓	×	0.04–0.11	collapse
5	✓	✓	×	0.410	-0.069

a different score level. The third property matters because the anchor mean is intended to approximate the report's baseline across omission severity. A batch concentrated at one rate estimates that rate's expected score, not the report-level offset. Table 2 reports five anchor schemes, each varying these three properties relative to the idealised anchor batch in scheme 1.

Scheme 1 uses the corpus's own controlled-omission summaries of the same report as anchors, holding out the summary under audit. It therefore matches the audited summary's model and prompt and spans the full omission-rate ladder. This idealised batch is the only scheme that recovers most of the offset, raising ρ from 0.479 to 0.580, or $+0.10$ out of the $+0.13$ oracle gap.

Schemes 2 and 3 keep rate spread but break one match at a time. In Scheme 2, the audited summary is fixed, and the same model is prompted to generate five additional anchor summaries of the same report with varying omission rates. This preserves the model and attempts to obtain a rate-spanning anchor batch without reconstructing the full controlled corpus, but the anchor-generation prompt differs from the standard summarisation prompt that produced the audited summary. The gain disappears ($\Delta\rho = -0.009$). Scheme 3 keeps the standard prompt and rate spread, but uses six anchors from the three alternative models at six randomly sampled distinct rates; it also fails to improve over the baseline ($\Delta\rho = -0.007$). Together, these results indicate that the offset is tied to the specific report–model–prompt combination that produced the audited summary.

The two no-spread schemes isolate the role of rate spread. Scheme 4 uses one to three anchors from the other models at the audited summary's own rate. With no rate spread, the anchor mean estimates the expected score at that rate rather than the report baseline; subtracting it removes much of the omission signal itself, collapsing ρ to approximately 0.04–0.11. Scheme 5 re-runs the same model and prompt on the full report while varying only the sampling seed; the resulting summaries are faithful and near-complete, so the anchor batch has no omission-rate spread. Despite matching the model and prompt, it hurts performance ($\Delta\rho = -0.069$), because the anchor mean sits near the low-omission end rather than near the report's average score over omission levels.

Table 3: Additional Negative Levers Relative to the Tuned Baseline ($\rho = 0.479$)

lever	intervention	$\Delta\rho$
2	salience-weighted aggregation	≤ -0.04
3	isotropy correction	$\leq +0.01$
4	additional domain-adaptive tuning	≈ 0.00
5	FTW2V-to-transformer bridge	-0.264
6	fact-level density estimation	-0.262

The conclusion is that offset removal is possible only with an anchor batch that matches the audited model and prompt and spans omission rates. This condition conflicts with deployment. Faithful resampling of a report produces mostly complete summaries, not a rate-spanning anchor batch. Reproducing scheme 1 would require segmenting the report into facts, deleting varying fractions, and regenerating summaries with the same model and prompt, which amounts to reconstructing the controlled-omission pipeline for every audited report. Lever 1 is therefore an upper-bound result: the offset is removable in principle, but not under the intended frugal deployment constraint.

5.3 Five additional levers

Beyond anchors, five additional interventions targeted either the per-report offset or the signal-to-noise ratio of the omission score. Table 3 summarises their effect relative to the tuned baseline ($\rho = 0.479$).

The only neutral intervention is additional domain-adaptive tuning; all others either fail to improve the score or degrade it substantially. Together with Lever 1, these results support the same conclusion: under the frugal single-pair constraint, the main remaining bottleneck is the report-specific offset rather than an easily replaceable component of the pipeline.

6 Comparison to No-Gold-Summary Metrics and Cost

To place the method in the wider field, we re-evaluated a representative set of no-gold-summary metrics on the same held-out split, under the same on-premise constraint. Each metric is used as a

per-pair omission signal, and all scores are oriented so that higher means more omission. The comparison covers five families: frugal string baselines (length ratio and ROUGE recall), static and contextual embedding coverage (FTW2V coverage; BERTScore (Zhang et al., 2020) with DrBERT (Labrak et al., 2023) and CamemBERT-bio (Touchent and Villemonte de la Clergerie, 2024; Martin et al., 2020)), NLI source coverage with mDeBERTa (He et al., 2021) in the style of SummaC (Laban et al., 2022), and three LLM-based metrics computed with local open-weight models served on-premise: GPTScore (Fu et al., 2024), G-Eval (Liu et al., 2023), and ChainPoll (Friel and Sanyal, 2023).

Each LLM metric is run with three open-weight models: Gemma 4 31B (Google DeepMind, 2026), Qwen3 30B (Yang et al., 2025), and Mistral Small 24B (Mistral AI, 2025). Gemma 4 31B also generated the main corpus summaries, while the other two models did not; this provides a circularity control, since a judge based on the generating model may favour outputs produced by the same model. Because the original EmbedKDECheck baselines used GPT-family models, the present comparison should not be read as a replication of those absolute figures. Instead, it compares metric families under the on-premise, open-weight constraint of this study.

Statistical comparisons use the paired clustered bootstrap described in Section 4.2. For each method, we compute $\Delta\rho = \rho(\text{EmbedKDECheck}) - \rho(\text{method})$ on the same resampled test reports, and report the 95% confidence interval together with $P = P(\Delta\rho > 0)$, the fraction of bootstrap resamples in which EmbedKDECheck has the higher rank correlation. We call EmbedKDECheck significantly higher when $P \geq 0.975$, the competing method significantly higher when $P \leq 0.025$, and otherwise report a tie.

The results are summarised in Table 4; the cost–quality trade-off is shown in Appendix D. EmbedKDECheck significantly outranks every frugal, static, contextual, and entailment metric on graded rank correlation, with paired $\Delta\rho$ from $+0.17$ to $+0.27$ and $P = 1.00$ in all these comparisons. Some frugal baselines achieve comparable or slightly higher thresholded F1 values, but their much lower ρ and AUC@50 show that they do not rank omission severity reliably. This reinforces the protocol choice of treating ρ and AUC@50 as the primary discrimination metrics.

Among the LLM metrics, only G-Eval significantly outperforms EmbedKDECheck on graded rank correlation: the two independent judges reach $\rho \approx 0.61$, while the generating-model judge reaches $\rho = 0.717$. However, this advantage is metric-specific and costly. The independent G-Eval judges tie EmbedKDECheck on AUC@50 and F1@50, and the gain in graded ranking requires 10^7–10^8 times more FLOPs. GPTScore and ChainPoll either tie or underperform the frugal pipeline on ρ, depending on the judging model.

On cost, the contrast is large enough to change the deployment interpretation. The reducer choice alone lowers the static pipeline cost by roughly a factor of six because random projection avoids a per-pair SVD fit. The resulting FTW2V+RP configuration lies on the ρ–cost frontier: it is significantly exceeded in rank correlation only by G-Eval-style LLM judges, which are 10^7 to 10^8 times more expensive in FLOPs and require 46–62 GB of peak memory, compared with 5.8 GB for EmbedKDECheck. For a screening tool intended to run on every generated summary, this is the difference between a lightweight background process and a multi-GPU inference budget.

7 Discussion and Conclusion

The main practical finding is the shape of Lever 1. With a single summary arriving alone, the per-report offset cannot be removed within the intended frugality budget. The operating point is therefore the tuned FTW2V+random-projection detector: $\rho \approx 0.48$ for graded omission ranking and F1@50 ≈ 0.80 for complete-versus-half-omitted screening. The bottleneck is not primarily the reducer, aggregation rule, bandwidth, or embedding family within the tested design space; it is the report-specific score offset. This means that further gains are unlikely to come from another small local tuning change unless that change also calibrates scores across reports.

A practical strength of the density-ratio formulation is that its score is inspectable. The document score aggregates per-token plausibilities, so a flagged summary can be decomposed into source tokens or spans that fell in low-density regions of the summary cloud. These tokens are not guaranteed omissions, but they provide an auditable starting point: the system can show which source content it considers poorly covered rather than re-

Table 4: Comparison with no-gold-summary metrics on the held-out split. $\Delta\rho = \rho(\mathrm{EKC}) - \rho(\mathrm{method})$.

family	method	ρ	F1@50	AUC@50	mean F1	$\Delta\rho$ vs EKC [95% CI]	EKC wins on ρ	TFLOPs/pair	RAM
this work	**EmbedKDECheck (FTW2V+RP)**	**0.479**	**0.796**	**0.881**	**0.656**	(reference)	(reference)	5.7×10^{-7}	5.8
frugal	length ratio	0.227	0.803	0.693	0.671	$+0.252$ $[+0.20, +0.30]$	**yes**	4.1×10^{-10}	0.1
frugal	ROUGE recall	0.256	0.797	0.708	0.686	$+0.224$ $[+0.16, +0.28]$	**yes**	1.5×10^{-9}	0.1
static	FTW2V coverage	0.307	0.770	0.737	0.662	$+0.173$ $[+0.12, +0.23]$	**yes**	1.9×10^{-6}	5.8
contextual	BERTScore (DrBERT)	0.209	0.745	0.680	0.657	$+0.270$ $[+0.19, +0.34]$	**yes**	0.215	0.5
contextual	BERTScore (CamemBERT-bio)	0.277	0.769	0.728	0.671	$+0.202$ $[+0.12, +0.28]$	**yes**	0.215	0.5
entailment	NLI (mDeBERTa)	0.238	0.712	0.685	0.652	$+0.241$ $[+0.16, +0.31]$	**yes**	1.2	1.3
GPTScore	Gemma 4 31B*	0.257	0.648	0.681	0.547	$+0.223$ $[+0.12, +0.32]$	**yes**	55.1	62.3
GPTScore	Qwen3 30B	0.449	0.723	0.770	0.635	$+0.031$ $[-0.08, +0.15]$	tie	6.5	59
GPTScore	Mistral Small 24B	0.564	0.796	0.876	0.712	-0.084 $[-0.17, +0.01]$	tie	42.9	45.9
G-Eval	Gemma 4 31B*	**0.717**	0.937	0.975	0.839	-0.237 $[-0.33, -0.14]$	no	63.0	62.3
G-Eval	Qwen3 30B	0.611	0.819	0.891	0.738	-0.131 $[-0.23, -0.04]$	no	7.2	59
G-Eval	Mistral Small 24B	0.606	0.814	0.888	0.747	-0.127 $[-0.22, -0.05]$	no	56.3	45.9
ChainPoll	Gemma 4 31B*	0.422	0.896	0.941	0.762	$+0.058$ $[-0.02, +0.14]$	tie	171.2	62.3
ChainPoll	Qwen3 30B	0.186	0.704	0.763	0.561	$+0.293$ $[+0.24, +0.35]$	**yes**	20.7	59
ChainPoll	Mistral Small 24B	0.104	0.706	0.703	0.566	$+0.375$ $[+0.32, +0.43]$	**yes**	141.4	45.9

* model that also generated the main corpus summaries; Qwen3 30B and Mistral Small 24B are independent judges. RAM is in GB.

turning only an opaque scalar score. For clinical summarisation, this makes the method useful not as a final arbiter, but as a cheap on-premise guard rail for surfacing summaries whose source coverage deserves human review.

Limitations

Several limitations bound the conclusions. First, the main detection results are reported for one summariser and one document type: Gemma 4 31B summaries of French synthetic clinical reports. The corpus contains summaries from three additional open-weight models, so cross-summariser validation is a natural next step. Second, the omission construct is controlled fact deletion before summarisation. This gives a known, monotonic omission driver, but it does not cover subtler real-world omissions such as hedging, de-emphasis, loss of certainty, or omission of clinically important context without removing an entire atomic fact. Future corpora should therefore include importance-weighted facts and expert-annotated naturally occurring omissions.

Third, the controlled rates depend on automatic fact segmentation. We mitigate this by selecting the segmenter empirically for high coverage, faithfulness, and fine granularity, and by evaluating against achieved rather than target deletion rates. Still, the resulting fact inventories were not independently validated for semantic atomicity, so segmentation errors may introduce label noise. Fourth, the binary F1@50 reported here is measured on a graded controlled-deletion corpus and should not be compared directly with the original EmbedKDECheck binary benchmark; the task, summariser set, and corpus construction differ. In addition, the synthetic reports were split by report rather than by their 50 underlying seed reports because seed provenance was not available in our working corpus; since EmbedKDECheck is training-free, this concern is limited to hyperparameter selection and threshold calibration rather than detector training, but a seed-disjoint split would provide a stricter validation. Finally, some mechanistic interpretations remain hypotheses. In particular, the static-to-contextual bridge appears limited by data scale, and fact-level density estimation may suffer from sparse pooled fact clouds, but these explanations were not isolated by dedicated controls. They should be treated as directions for follow-up rather than as established mechanisms.

Code and Data Availability

The controlled-omission benchmark is publicly available on Hugging Face at https://huggingface.co/datasets/Hodhaifa/fr-clinical-graded-omission. The code and experimental configurations used in this study are publicly available at https://github.com/BenouaklilHodhaifa/frugal-graded-omission-inlg2026. The corpus consists of fictitious synthetic clinical reports inspired by anonymized medical reports and does not contain identifiable patient records.

Ethics Statement

This work uses fictitious synthetic French clinical reports inspired by previously anonymized medical reports, following the construction described in the original EmbedKDECheck paper (Oukelmoun et al., 2025a). We do not process identifiable patient records in the present study. The motivation for using synthetic clinical text is to avoid exposing personal health data while studying omission-

detection methods relevant to privacy-constrained medical settings. The detector itself requires no external LLM or API call and can therefore operate without transmitting clinical text to an external evaluator; privacy of the overall system still depends on how the upstream summariser is deployed. The method is intended as a screening aid for human review, not as an autonomous clinical decision system.

References

Josh Achiam et al. 2023. GPT-4 technical report. arXiv:2303.08774.

Dimitris Achlioptas. 2003. Database-friendly random projections: Johnson–Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687.

Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. 2001. On the surprising behavior of distance metrics in high dimensional space. In *Proceedings of the International Conference on Database Theory*.

Hodhaifa Benouaklil, Achir Oukelmoun, Nasredine Semmar, Faiza Belbachir, Jaafer Klila, and Hassane Essafi. 2026. fr-clinical-graded-omission: A graded omission benchmark for french clinical summarisation. https://huggingface.co/datasets/Hodhaifa/fr-clinical-graded-omission.

Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. 1999. When is nearest neighbor meaningful? In *Proceedings of the International Conference on Database Theory*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Abhimanyu Dubey et al. 2024. The Llama 3 herd of models. arXiv:2407.21783.

European Parliament and Council of the European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data (General Data Protection Regulation). Official Journal of the European Union, L119, 1–88.

Alexander R. Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved QA-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

Robert Friel and Atindriyo Sanyal. 2023. ChainPoll: A high efficacy method for LLM hallucination detection. arXiv:2310.18344.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. GPTScore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 6556–6576.

Google DeepMind. 2026. Gemma 4 Model Card. [Online]. Available: https://ai.google.dev/gemma/docs/core/model_card_4. Accessed: Jun. 29, 2026.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *Proceedings of the International Conference on Learning Representations*.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10.

Yanis Labrak, Adrien Bazoge, Richard Dufour, Mickael Rouvier, Emmanuel Morin, Béatrice Daille, and Pierre-Antoine Gourraud. 2023. DrBERT: A robust pre-trained model in french for biomedical and clinical domains. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2024. Factual consistency evaluation of summarisation in the era of large language models. *Expert Systems with Applications*.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017.

Louis Martin et al. 2020. CamemBERT: A tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219.

Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv:1802.03426.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. arXiv:1301.3781.

Mistral AI. 2024. Un ministral, des ministraux (Ministral 8B). [Online]. Available: https://mistral.ai/news/ministraux/. Accessed: Jun. 29, 2026.

Mistral AI. 2025. Mistral Small 3. [Online]. Available: https://mistral.ai/news/mistral-small-3/. Accessed: Jun. 29, 2026.

Achir Oukelmoun, Nasredine Semmar, Gaël de Chalendar, Clément Cormi, Mariame Oukelmoun, Eric Vibert, and Marc-Antoine Allard. 2025a. Detecting omissions in LLM-generated medical summaries. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 325–337.

Achir Oukelmoun, Nasredine Semmar, Gaël de Chalendar, Clément Cormi, Mariame Oukelmoun, Eric Vibert, and Marc-Antoine Allard. 2025b. Détection des omissions dans les résumés médicaux générés par les grands modèles de langue. In *Actes des 32ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : articles scientifiques originaux*, pages 252–267, Marseille, France. ATALA & ARIA.

Achir Oukelmoun, Nasredine Semmar, Gaël de Chalendar, Enguerrand Habran, Eric Vibert, Emma Goblet, Mariame Oukelmoun, and Marc-Antoine Allard. 2023. A study on the relevance of generic word embeddings for sentence classification in hepatic surgery. In *Proceedings of the 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*, pages 1–8.

République Française. 2026. Code de la santé publique, Article L1111-8: Hébergement des données de santé à caractère personnel. Légifrance. Version in force; accessed Jun. 29, 2026.

Bernard W. Silverman. 1986. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, U.K.

Rémi Touchent and Éric Villemonte de la Clergerie. 2024. CamemBERT-bio: Leveraging continual pretraining for cost-effective models on french biomedical data. In *Proceedings of LREC-COLING*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Stanisław Węglarczyk. 2018. Kernel density estimation and its application. In *ITM Web of Conferences*, volume 23, page 00037.

An Yang et al. 2025. Qwen3 technical report. arXiv:2505.09388.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *Proceedings of the International Conference on Learning Representations*.

A Segmenter Comparison

Table 5: Segmenter comparison over the 460 reports. Coverage is ROUGE-1 recall, faithfulness is ROUGE-1 precision.

model	coverage	faithfulness	mean facts	median facts
Llama 3 70B	$\mathbf{0.991} \pm 0.021$	0.999 ± 0.002	42.3	31.5
Llama 3.1 8B	0.976 ± 0.055	0.993 ± 0.027	39.8	35.0
Mistral Small 24B	0.989 ± 0.017	1.000 ± 0.002	43.4	34.0
Ministral 8B	0.977 ± 0.058	0.998 ± 0.009	33.4	29.0
Qwen3 30B	0.988 ± 0.018	0.999 ± 0.002	39.8	31.0
Gemma 4 31B	$0.990 \pm \mathbf{0.014}$	$\mathbf{1.000 \pm 0.000}$	**46.2**	**37.0**

B Prompts

This appendix lists the prompts used for corpus construction and anchor generation. The French prompts are the prompts actually used in the experiments. English translations are provided only for readability.

B.1 Dataset-generation prompts

Clinical summary prompt. Used to generate the main clinical summaries.

```
Tu es un médecin expérimenté. Rédige un résumé clinique du compte rendu
médical ci-dessous. Le résumé doit être fidèle, couvrir toutes les
informations cliniquement importantes, et tenir en 8 phrases maximum et
200 mots au total. N'ajoute AUCUNE information absente du texte.
Réponds uniquement avec le résumé, sans commentaire.
```

English translation. Translation for readability only; the French prompt above is the one used in the experiments.

```
You are an experienced physician. Write a clinical summary of the
medical report below. The summary must be faithful, cover all
clinically important information, and fit within 8 sentences and 200
words in total. Do not add ANY information absent from the text. Answer
only with the summary, without commentary.
```

Report segmentation prompt. Used to decompose reports into atomic facts.

```
Tu es un expert en structuration de comptes rendus médicaux en français.

Tâche :
Transformer le compte rendu en une liste numérotée d'éléments, sans
perdre AUCUN contenu du texte source.

Règles obligatoires :

Conservation totale
Reproduis absolument tout le contenu du compte rendu.
N'omets aucun mot, aucun titre, aucun sous-titre, aucune puce, aucune
information.
Ne résume pas. Ne reformule pas. Ne corrige pas.

Détection des titres et sous-titres
Tout segment ayant une fonction de rubrique ou d'organisation doit être
traité comme un titre, même si sa forme varie.
Appuie-toi sur le contexte et le rôle du segment dans le document, pas
seulement sur la mise en page.
Un titre peut être : en majuscules ou non, avec ou sans ":", précédé de
"·", "-", "*" ou autre, seul sur une ligne, très court.
Tous les titres et sous-titres doivent être recopiés à leur place
d'origine et préfixés par exactement : #
```

```
Segmentation des faits
Chaque fait doit être dans une entrée distincte (un fait = une seule
information principale).
Si une ligne contient un titre suivi de contenu, sépare le titre en une
entrée, puis les faits en entrées séparées.

Priorité en cas de doute
Priorité 1 : ne rien perdre.
Priorité 2 : préserver la structure.
Priorité 3 : si un segment semble organiser les informations suivantes,
traite-le comme un titre.

Erreurs interdites :

* oublier un sous-titre parce qu'il n'est pas en majuscules ou n'a pas
de ":" ;
* transformer un titre en fait ou le fusionner avec son contenu ;
* supprimer une ligne courte ;
* reformuler ou regrouper plusieurs faits en une seule entrée.

Format de sortie :
Réponds uniquement en français.
Commence directement par "1." sans aucune phrase d'introduction ni de
conclusion.
Chaque élément occupe exactement une ligne numérotée.
Les titres et sous-titres commencent par "# ".
Formate ta réponse EXACTEMENT ainsi :

1. # <intitulé de section>
2. <premier fait>
3. <deuxième fait>
4. # <autre intitulé>
5. <fait suivant>

... et ainsi de suite. Aucun texte supplémentaire, aucun commentaire.
```

English translation. Translation for readability only; the French prompt above is the one used in the experiments.

```
You are an expert in structuring French medical reports.

Task:
Transform the report into a numbered list of elements, without losing
ANY content from the source text.

Mandatory rules:

Total preservation
Reproduce absolutely all the content of the report.
Do not omit any word, title, subtitle, bullet point, or information.
Do not summarise. Do not rephrase. Do not correct.

Detection of titles and subtitles
Any segment that has a sectioning or organisational function must be
treated as a title, even if its form varies.
Rely on the context and the role of the segment in the document, not
only on formatting.
A title may be uppercase or not, with or without ":", preceded by "•",
"-", "*" or another marker, alone on a line, or very short.
All titles and subtitles must be copied in their original position and
prefixed exactly with: #

Fact segmentation
Each fact must be in a separate entry (one fact = one main piece of
information).
If a line contains a title followed by content, separate the title into
one entry, then separate the facts into distinct entries.

Priority in case of doubt
Priority 1: lose nothing.
Priority 2: preserve the structure.
Priority 3: if a segment seems to organise the following information,
treat it as a title.

Forbidden errors:

* forgetting a subtitle because it is not uppercase or does not have
":";
* turning a title into a fact or merging it with its content;
* deleting a short line;
* rephrasing or grouping several facts into one entry.

Output format:
Answer only in French.
Start directly with "1." without any introductory or concluding
sentence.
Each item occupies exactly one numbered line.
Titles and subtitles begin with "# ".
Format your answer EXACTLY as follows:

1. # <section heading>
2. <first fact>
3. <second fact>
4. # <other heading>
5. <next fact>

... and so on. No additional text, no commentary.
```

Multi-omission anchor prompt. Used to generate five post-hoc anchor summaries with increasing omission levels.

```
Tu es un médecin expérimenté. À partir du compte rendu médical
ci-dessous, rédige CINQ résumés cliniques du MÊME compte rendu, chacun
omettant volontairement une proportion CROISSANTE des faits cliniques.

Proportions cibles de faits à omettre (par rapport à l'ensemble des
faits du compte rendu) :

* NIVEAU 1 : environ 10 %
* NIVEAU 2 : environ 25 %
* NIVEAU 3 : environ 40 %
* NIVEAU 4 : environ 55 %
* NIVEAU 5 : environ 70 %

Règles obligatoires :

* Pour chaque niveau, tu décides toi-même de la proportion EXACTE de
faits que tu omets, en restant proche de la cible.
* Omets des faits ENTIERS (une information clinique complète), pas
seulement des mots : plus le niveau est élevé, moins le résumé contient
d'informations.
* Chaque résumé doit rester un résumé clinique fidèle des faits
CONSERVÉS. N'ajoute AUCUNE information absente du compte rendu,
n'invente rien.
* Rédige chaque résumé en prose continue, sans titres internes ni
listes à puces.
* Indique pour chaque niveau le pourcentage approximatif de faits que
tu as RÉELLEMENT omis.

Format de sortie EXACT (aucune introduction, aucune conclusion, rien
d'autre) :

### NIVEAU 1 | omission: <pourcentage>%

<résumé du niveau 1>

### NIVEAU 2 | omission: <pourcentage>%

<résumé du niveau 2>

### NIVEAU 3 | omission: <pourcentage>%

<résumé du niveau 3>

### NIVEAU 4 | omission: <pourcentage>%

<résumé du niveau 4>

### NIVEAU 5 | omission: <pourcentage>%

<résumé du niveau 5>
```

English translation. Translation for readability only; the French prompt above is the one used in the experiments.

```
You are an experienced physician. From the medical report below, write
FIVE clinical summaries of the SAME report, each deliberately omitting
an INCREASING proportion of clinical facts.

Target proportions of facts to omit, relative to all facts in the
report:

* LEVEL 1: about 10%
* LEVEL 2: about 25%
* LEVEL 3: about 40%
* LEVEL 4: about 55%
* LEVEL 5: about 70%

Mandatory rules:

* For each level, you decide yourself the EXACT proportion of facts to
omit, while staying close to the target.
* Omit WHOLE facts, not only words: the higher the level, the less
information the summary contains.
* Each summary must remain a faithful clinical summary of the facts
that are KEPT. Do not add ANY information absent from the report,
invent nothing.
* Write each summary in continuous prose, without internal titles or
bullet lists.
* For each level, indicate the approximate percentage of facts that you
ACTUALLY omitted.
```

```
EXACT output format (no introduction, no conclusion, nothing else):

### LEVEL 1 | omission: <percentage>%

<level 1 summary>

### LEVEL 2 | omission: <percentage>%

<level 2 summary>

### LEVEL 3 | omission: <percentage>%

<level 3 summary>

### LEVEL 4 | omission: <percentage>%

<level 4 summary>

### LEVEL 5 | omission: <percentage>%

<level 5 summary>
```

C Reducer Geometry Diagnostics

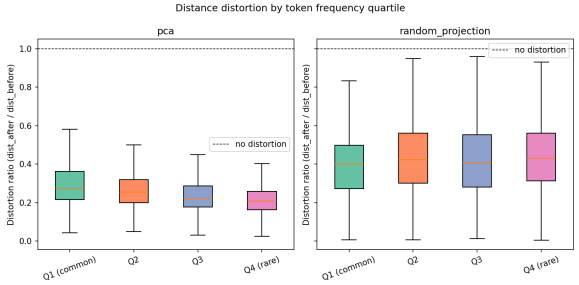


Figure 4: Distance distortion by token-frequency quartile. PCA compresses rare-token distances more strongly than frequent-token distances, whereas random projection distorts frequency bands more evenly.

D Efficiency Frontier

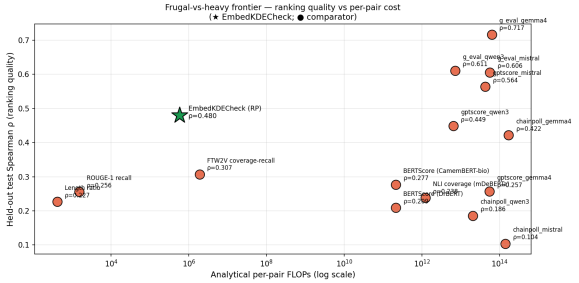


Figure 5: Efficiency frontier: held-out Spearman ρ against per-pair floating-point operations on a log scale. EmbedKDECheck lies at the low-cost edge; higher rank correlation is achieved only by LLM judges several orders of magnitude to the right.